CHEMO: Collaborative Chemoinformatics Open Platform

Camilo Ramírez-Sánchez1, Jessica Valero-Rojas2,3, Claudia Martínez-García4, Javier Eduardo Jaimes1, Paula A. Santana5, Alexis Salas2, David Ramírez2,\*.

1 Institución Universitaria Politécnico Grancolombiano. Colombia.

2 Departamento de Farmacología, Universidad de Concepción, Chile

3 Doctorado en Ciencias Químicas, Facultad de Ciencias Químicas, Universidad de Concepción, Chile.

4 Departamento de Farmacia, Facultad de Ciencias, Universidad Nacional de Colombia, Colombia

5 Facultad de Ingeniería, Instituto de Ciencias Químicas Aplicadas, Universidad Autónoma de Chile, Chile.

Corresponding author: David Ramírez - [dramirezs@udec.cl](mailto:dramirezs@udec.cl)

# ABSTRACT

# GRAPHICAL ABSTRACT

# KEYWORDS

# INTRODUCTION

Nowadays chemoinformatics tools are crucial for drug development pipelines and provides to researchers the ability to identify potential drug candidates and optimize their pharmacological properties through the use of computational methods (Giordanetto & Kihlberg, 2014; Verma & Khedkar, 2016; Kumar & Chandra, 2019; Bohacek et al., 1996). However, the effective teaching/learning of chemoinformatics as well as other computer-aided drug design (CADD) tools is challenging. To achieve this, researchers and students who do not have a background in programming or computer science requires specialized knowledge and skills that are often difficult to acquire through traditional classroom instruction alone (Guvench, 2015). To tackle this challenge, a number of web-based teaching platforms have been developed, in order to provide students with the opportunity to learn key concepts and skills in a more interactive and engaging way. For example, MolSim offers a molecular simulation platform for learning (Wang et al., 2018). PharmMapper, is a web server for potential drug target identification using a pharmacophore mapping approach (Zhao et al., 2012), and TeachOpenCADD is a teaching platform for CADD that uses open source packages and data (Sydow et al., 2019). These platforms offer a range of modules and tools for teaching ligand- and structure-based drug design, virtual screening, and ADME-Tox prediction, among other topics, and many of them are based on open source packages, enabling students to learn chemoinformatics and CADD using accessible and affordable tools.

METER MODELO EDUCATIVO de la UdeC

In this work, we present CHEMO (collaborative CHEMoinformatics Open platform) a web-based teaching platform for chemoinformatics that uses Python and various scientific packages, such as NumPy, Pandas, and Matplotlib, to teach key concepts and skills in drug design and molecular modeling. The platform is designed to be accessible to a wide range of students, from beginners to advanced, and covers a range of topics, including an introduction to Python, data manipulation and visualization, virtual screening and machine learning in drug design. The curriculum is designed to be modular, allowing instructors to adapt the material to their specific needs and teaching methods.

Our CHEMO Platform is building on previously established platforms, such as OpenDrugDiscovery (ODD), Cheminformatics and QSAR Society (CISS), and TeachOpenCADD. However, CHEMO stands apart due to its implementation in English and also in Spanish, since we consider that in countries where English is not the mother tongue, and where due to the lack of resources and development, university students have a very low level of English as a second language, it is necessary to provide tools in the native language that help to reduce the cultural barrier in order to be able to potentially improve the learning process. In Latin America there is a high rate of students with a low level of English proficiency, and having a significant number of students being trained in different disciplines in universities, it is necessary to provide them with tools to acquire new skills that allow them to face the challenges of drug design in a better way. It is also important to notice that CHEMO is implemented in Jupyter as an open-source web application, with Python and other open-source scientific packages that are widely used in the scientific community. Jupyter notebooks provide an interactive environment for students to write and run Python code directly within their web browser, making their learning experiences more interactive and engaging. This fits out students with a solid foundation for learning more advanced topics in CADD. Moreover, CHEMO takes on real-world examples and hands-on applications to increase student learning and engagement. The modular design of our curriculum also affords flexibility in the course design and delivery. A distinctive key factor of CHEMO is that researchers who may not be familiar with the programming language, can find a gentle and structured introduction to Python for drug design and modeling. Our goal is to examine the potential of CHEMO, our Collaborative Chemoinformatics Open Platform, and to demonstrate its versatility and power as a tool for teaching and learning drug design and modeling in Latin America. By outlining the key features and benefits of CHEMO, we aim to highlight its potential as a valuable resource for students of all levels.

# BACKGROUND

Jupyter is an open-source web application that allows users to create, share, and collaborate on documents that contain live code, equations, visualizations, and narrative text (Kluyver et al., 2016). One of the primary advantages of Jupyter is that it allows users to write and test code in the same environment as the documentation, making it a powerful tool for interactive computing and data analysis.

Python is a general-purpose programming language that is widely used in fields like machine learning, scientific computing, and web development (van Rossum & Drake, 1995). Python’s popularity can be attributed to its simple syntax, strong support for multiple programming paradigms, and its vast ecosystem of third-party packages and libraries. Nowadays, some programs used in chemoinformatics and drug design are based on python. That's the reason why students should know how to use it. They are going to understand how the program works and they are going to be able to solve any problems.

The combination of Jupyter and Python has become increasingly popular among researchers, educators, and data scientists due to the ease of creating and sharing interactive, reproducible workflows. In the context of computer-aided drug design (CADD), Jupyter and Python are particularly well-suited for visualizing, analyzing, and manipulating molecular data (Murray-Rust et al., 2019). Several popular Python libraries have emerged that are specifically geared towards CADD, including RDKit, Autodock, and PyMOL. These libraries provide functionality for tasks such as ligand-based and structure-based drug design, virtual screening, and molecular dynamics simulations (RDKit, n.d.; Trott & Olson, 2010; Čermák et al., 2017).

Computer-aided drug design is a set of techniques that allows the selección of new compounds that can be used as active compounds in drugs (Del Rio & Varchi, 2016). CADD selection is based on 4 phases; the first one is a virtual screening using different libraries of compounds against the diana molecule. The second one is to perform a molecular docking in order to evaluate the specificity of the molecules resulting from virtual screening. The third one, is to use *in silico* techniques in order to evaluate the absorption, distribution, metabolism, excretion, and toxicity (ADMET) of the molecules resulting from the second phase. The last phase consists in designing the drug based on structure or based on ligands (Arya & Coumar, 2021).

Computer-aided drug design is a multidisciplinary field that involves the combination of computational and experimental techniques to identify and optimize potential drug candidates (Ruiz-Carmona et al., 2014). The central goal of CADD is to speed up the drug-discovery process by reducing the time and cost associated with experimental synthesis and testing of compounds. The CADD workflow typically involves four phases, including virtual screening, molecular docking, *in silico* ADMET evaluation, and drug design based on structure or ligands (Arya & Coumar, 2021).

The use of Jupyter, Python, and scientific packages in CADD has enabled researchers and educators to take advantage of the interactive and collaborative features of Jupyter while leveraging the powerful computational tools and libraries available in the Python ecosystem. This has led to the development of numerous open-source tools and platforms for CADD which facilitate the design, optimization, and screening of potential drug candidates (Trott & Olson, 2010; Čermák et al., 2017). Our platform (CHEMO) provides tools to interact with databases such as CHEMBL, pubChem, among others, to download and analyze data, such as ADMET properties of a set of molecules, to perform hierarchical clustering of molecules with established similarities, and to build predictive models of bioactivity for a specific target. This allows researchers and scientists to design new drugs more efficiently and effectively, and obtain more precise and reliable results in their CADD research.

Several web-based teaching tools have been developed to enhance the teaching and learning of computer-aided drug design (CADD), such as AutoDock Vina, RDKit, and PyRx, among others. A literature review reveals that these tools generally fall into two categories: static web-based resources and dynamic web-based resources. Static web-based resources typically consist of text, images, and videos that provide explanations of various CADD concepts. These resources may be useful for gaining an initial understanding of CADD, but they often lack interactivity and engagement with the material. Dynamic web-based resources, on the other hand, allow for more interactive and engaging learning experiences. These resources may include simulations, virtual laboratories, and interactive tutorials that allow students to manipulate and visualize molecular data. Dynamic web-based resources have been shown to be particularly effective in improving student learning outcomes (Jensen et al., 2011).

One example of a dynamic web-based resource is the Virtual ChemLab, a software system developed by the company AD Instruments. The Virtual ChemLab allows students to perform virtual experiments in a simulated laboratory environment, with simulations available for a range of topics including CADD (AD Instruments, n.d.). Another example, DrugDesign@Home (Weiser & Saunders, 2018), is an online platform that allows students to perform virtual drug design experiments in real-time using open-source CADD software applications. However, there are limitations to these web-based resources. For example, some of these resources may lack proper integration of CADD-specific software and require the installation of software or tools on the user’s computer system, limiting accessibility and ease-of-use, especially for educators without technical knowledge.

Despite the potential limitations, web-based teaching tools have the potential to significantly improve the teaching and learning of CADD. By providing interactive, engaging, and accessible resources, these tools can provide students with a deeper understanding of CADD, and better prepare them for careers in the field.

# METHODS

The CHEMO consists of a series of interactive Jupyter notebooks that are organized into three main topics: Introduction to Python, Data manipulation and visualization, and Introduction to Machine Learning. These notebooks provide both a theoretical section with examples and code, as well as a practical section that covers the essential concepts in CADD.

The development of the platform for CADD was conducted in two phases. In the first phase, the team thoroughly identified the technical requirements and design specifications. They also provided detailed instructions on how to install and work with Conda. This allowed for the installation of specific package versions and dependencies, and the creation of virtual environments for individual modules. Moreover, a .yml file was provided that contained the necessary library container, making it simple and fast to install an environment with all the required packages.

To develop the platform, the team primarily used Jupyter, an open-source web application, and the programming language Python. To execute the platform's computational operations, several basic Python computing libraries were utilized, including numpy, pandas, matplotlib, seaborn, scikit-learn, requests, tarfile, and gzip (Pereira et al., 2020; Rosental et al., 2018). In addition to these libraries, the platform relied on several external resources, such as the ChEMBL, a database of bioactive molecules that contains information on their properties, targets, and activities(Gaulton et al., 2012; Koutsoukas et al., 2014), RDKit, an open-source software toolkit that provides functions for handling (RDKit, n.d.), analyzing, and visualizing chemical structures, PyPDB (Gilpin, 2015), a Python library that enables access to the Protein Data Bank (PDB) (Berman et al., 2000), which contains information on the 3D structures of proteins, nucleic acids, and other biomolecules, BioPandas (Raschka, 2017), a Python library that extends the functionality of the Pandas library for working with biological data, and PyMOL (Schrödinger, 2015; Čermák et al., 2017), a molecular visualization system that enables users to create high-quality images and animations of protein and small molecule structures.

In the second phase, the team focused on designing and developing the curriculum modules that would be part of the platform. The team carefully developed the curriculum to cover essential concepts in CADD, while also ensuring that the material was engaging, practical, and applicable in real-world scenarios. To create an effective curriculum, the team drew upon various resources, including published literature and feedback from experts in the field of CADD (Ma et al., 2015; Koutsoukas et al., 2014). They also included real-world examples and hands-on activities to help students apply the concepts they learned in the theoretical sections.

One of the primary design considerations was modularity. This means that the curriculum was structured in a way that allows instructors to adapt the material to their specific needs and teaching styles. For example, an instructor who wanted to focus more on virtual screening could easily do so by selecting and modifying the corresponding module in the curriculum. In addition to modularity, the team also emphasized user engagement through the use of interactive components in the Jupyter notebooks. For example, students were given the opportunity to manipulate and visualize data in real-time, enhancing their understanding of complex concepts in CADD.

The modules in CHEMO are divided into three parts: Introduction to the Python programming language (P1); where students work with basic Python concepts such as variables, arrays, formats, and control structures, along with three practices applied to CADD. Dataframes and visualization (P2); wherein libraries such as numpy and pandas are used to organize data in Dataframes, and libraries like matplotlib and seaborn are used to visualize data sets in different ways, with two CADD-specific practices included.

The third part focuses on Machine Learning (P3), introducing basic concepts and providing a practice orientation for the training of a ligand classification model for a specific target. The pipeline of modules and practices can be adapted to other examples by changing data such as the target protein or gene being studied. This design approach allows students and instructors to adjust the material to their needs and interests, resulting in a more personalized and effective learning experience in CADD.

# Results

In the following, the content of each module is briefly discussed, with a special emphasis on the practices of CADD.

## P1. Introduction to the Python programming language

This module (P1-M1) acts as a comprehensive guide to the Python programming language, highlighting CADD practices. Firstly, learners will be introduced to the programming language, and examples of native data types such as data identification, operation and manipulation. Learners will also gain insights on basic control structures, inclusive of conditionals and iterations.

As part of the practical section of the module, learners will delve into the manipulation of the protein enzyme, Cytochrome P450, which is integral to drug metabolism. Additionally, the module will also cover the chemical synthesis of lipids such as cholesterol and steroids. During this module, learners will perform the transcription of DNA sequences to RNA and extract amino acids responsible for encoding the protein, while gaining an understanding of their physical and chemical properties.

One of the key aspects of this module is to create a high level of perplexity and burstiness. Therefore, learners will be challenged to draw information on the number of polar, nonpolar, acidic and basic nucleotides present in the protein like the one shown in Figure X.

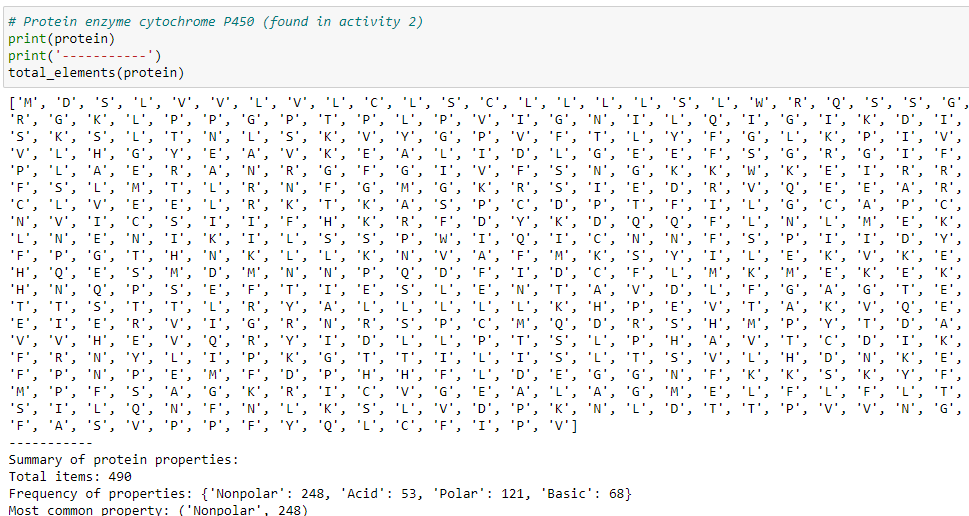


Figure X. Summary of protein enzyme, Cytochrome P450 properties and most common.

## P2. Dataframes and visualization

CHEMO platform’s P2 module comprises four distinct sub-modules and covers Dataframes and visualization. The primary focus of P2-M1 is to introduce learners to Dataframes as a fundamental structure to effectively organize and manipulate databases. In the practice session, learners will acquire information on compounds targeting the Glycogen Synthase Kinase-3 beta. The learners will be provided with the uniprot id of P49841, which further guides them to create a database containing 2657 molecules associated with the target P49841, along with their pchem\_values, SMILE representations, and 2D representations. The resulting database is then exported as a comma-separated document and referred to in subsequent modules. The first entries of the exported database are shown in Figure X.

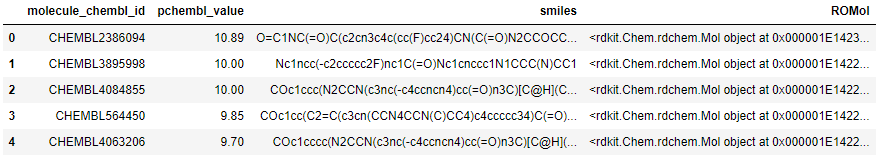


Figure X. Bioactivity data of the target P49841 and smiles, ROMol representations.

The second and third sub-modules (P2-M2, P2-M3) of CHEMO platform focus on introducing learners to the Matplotlib and Seaborn libraries for creating various types of graphical representations, such as 2D and 3D scatter plots, bar graphs, and correlation plots (P2-M2). In the practical session of P2-M3, learners will work with the database of compounds associated with the target P49841. They will filter compounds that can be used for drug development and apply the Lipinsky Rule of Five (Ro5) test to each of them. Out of 2657 initial molecules, 2477 molecules passed the test. The results will be presented in various forms such as bar graphs or pair plots, which will correlate the ADME properties of the compounds with the Ro5 test results. Moreover, the learners will use radar plots to visualize and export the ADME properties dataset for the next module as shown in Figure X.

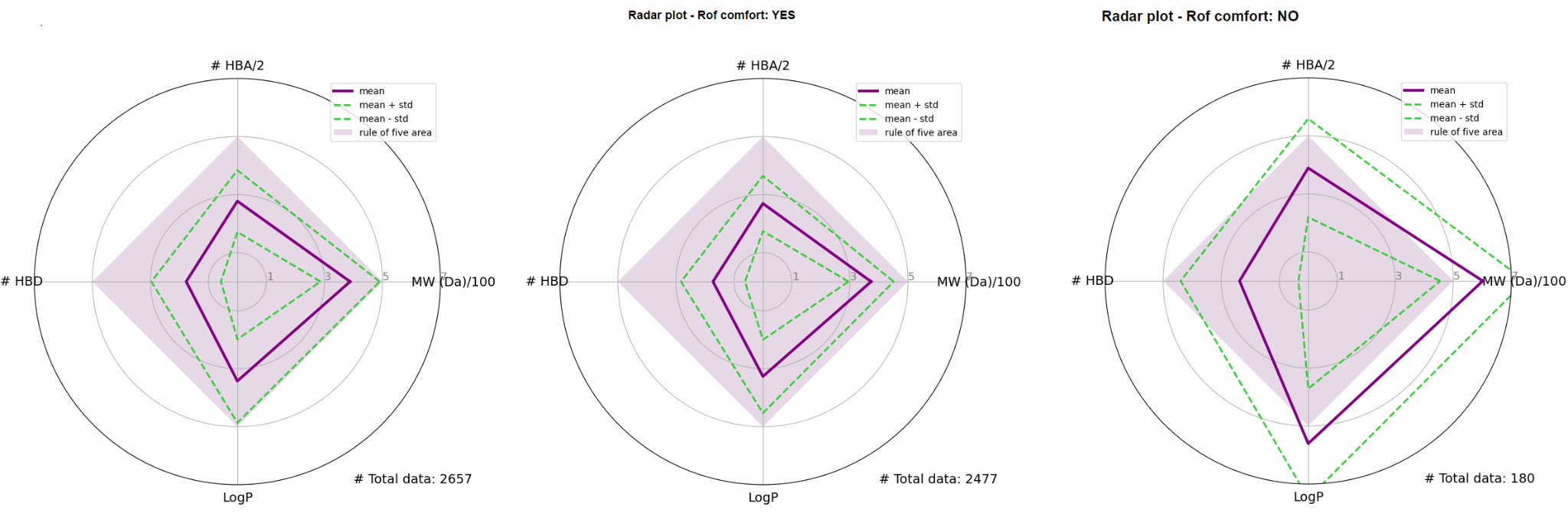


Figure X. Radar plot of ADME (absorption, distribution, metabolism, and excretion) properties of compounds that comply and do not comply with Rule of Five (Ro5)

The final sub-module of CHEMO platform (P2-M4) focuses on clustering the 2477 molecules into groups with similar effects. The primary objective of this module is to showcase practical applications of clustering techniques that can be employed in CADD research. The module involves finding the fingerprint of each compound using its SMILE representation and calculating the similarity between each molecule and the rest of the dataset using the Tanimoto distance. The results of the clustering process are presented visually using bar graphs, density plots, heat maps, and other data tables. For instance, one of the visual representations used in this module is a hierarchical clustering heat map, which arranges the similarity matrix shown in Figure X.

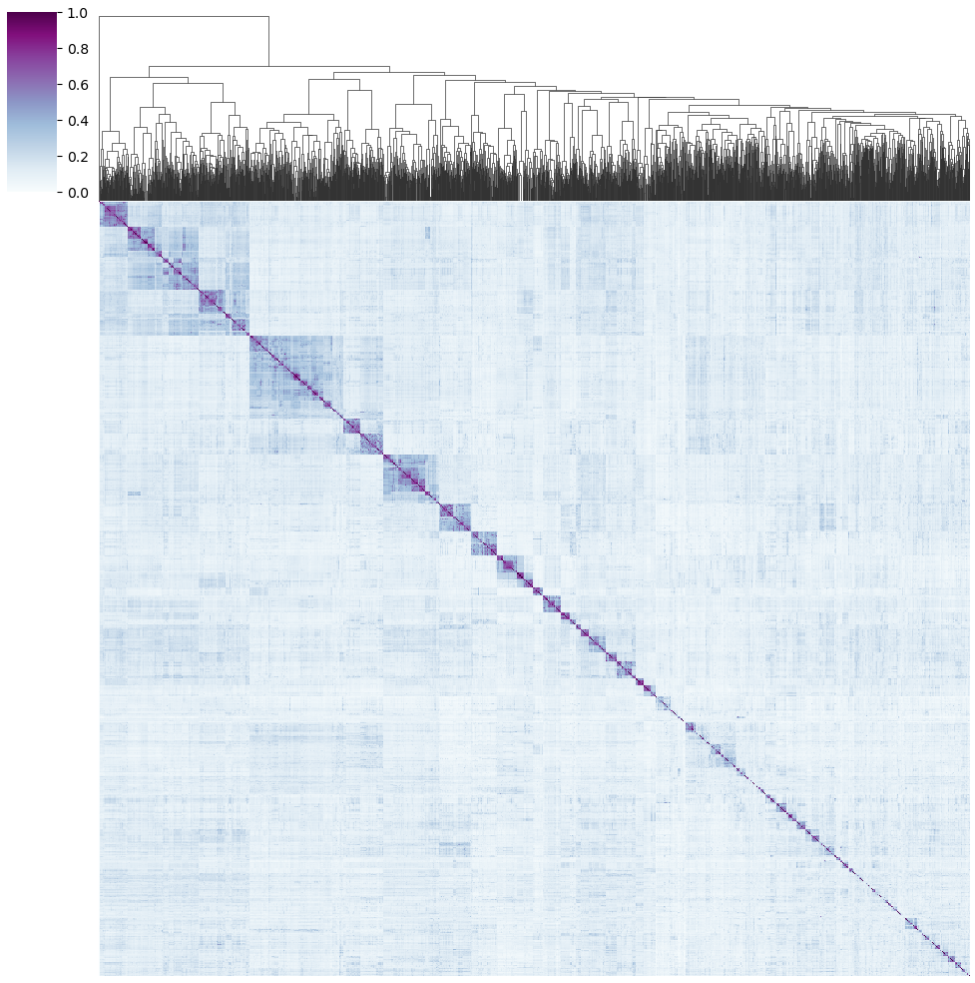


Figure X. Clustermap of the Tanimoto similarity matrix in a heat map of hierarchical clustering, where we can observe how molecules with the highest similarity are grouped.

Upon completion of the P2 module, learners will gain insights into the structure-functional relationships between compounds and identify novel drug candidates using clustering techniques. Thus, this module equips learners with advanced knowledge and practical skills required to utilize clustering techniques effectively in CADD research. Finally, the results obtained from the clustering process can provide insights into the structure-functional relationships between compounds and help in designing more effective and efficient drugs.

## P3. Introduction to Machine Learning

The third part of the CHEMO platform focuses on introducing learners to Machine Learning concepts and practical training. This part is divided into two modules: P3-M1, which serves as an introduction to ML concepts, and P3-M2, which involves practical training where an ML algorithm is trained to classify compounds. P3-M1 introduces learners to fundamental ML concepts such as the different types of learning, classification and regression models, and the Random Forest classification model, which is commonly used for its practicality. The introduction also includes various performance metrics used to evaluate ML models.

In the practical module P3-M2, students utilize a Random Forest classification algorithm that applies to the ligand database and ADME properties dataset from P2-M3. The Extended Connectivity Fingerprint (ECFP) representation, specifically ECFP4, is used as the list of features. Based on their pchem\_value, the compounds are divided into three categories: active, inactive, and intermediate, with the intermediate category being dropped as it is a binary classification problem. The initial dataset is divided into 70% for training and 30% for validation. The model’s effectiveness is measured using several performance metrics, including accuracy (accuracy train set = 100%, accuracy valid set = 88%), confusion matrix, ROC curve, and mean AUC value as shown in figure X. However, the model appears to overfit, and it is important either to optimize hyperparameters or use alternative classification models

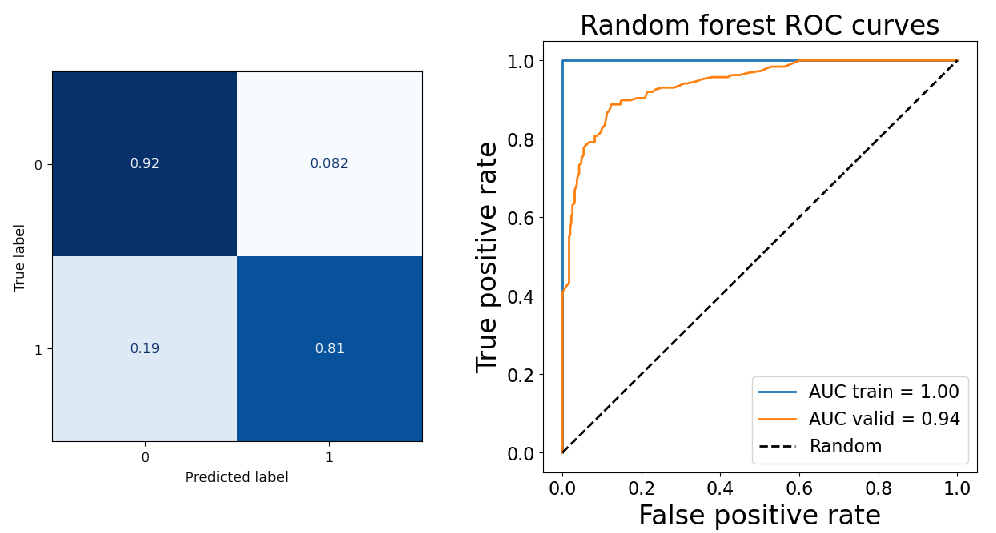


Figure X. Confusion matrix normalized and ROC curve for default model. The results indicate that the model is overfitting.

Consequently, a research team trained an XGBoost classification model for the same target and used 5-fold cross-validation to reduce overfitting of the Random Forest model. Using the same performance metrics, Table X compares the two models and highlights the efficiency of the optimized model over the default one and Figure X shows the confusion matrix and ROC curve of the optimized model.

**Table X. Comparison of metrics between default and optimized models**

| **Metrics** | **Random Forest Classifier (default)** | **XGBosst Classifier (optimized)** |
| --- | --- | --- |
| **Mean AUC** | 0.952 | 0.97 |
| **Accuracy** | 0.803 | 0.881 |
| **Sensitivity** | 0.896 | 0.933 |
| **Specificity** | 0.945 | 0.948 |

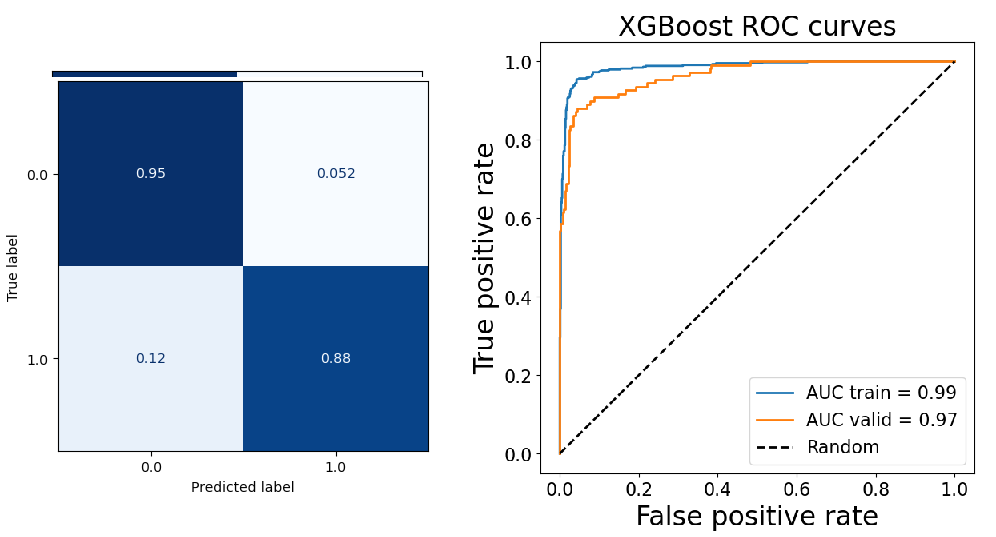


Figure X. Confusion matrix normalized and ROC curve for optimized model.

The final step of the CHEMO platform involves conducting a search for ten molecules with activity on the target that were not part of the training dataset. After this, predictions were made using both Random Forest (default) and XGBoost (optimized) models.To evaluate the model’s efficacy, Predictions were then conducted using both the default Random Forest and optimized XGBoost models. 10 randomly selected molecules from the validation set were used to compare the predictions. Table X presents the results, which demonstrate that the optimized model offers superior predictions in contrast to the default Random Forest model. These outcomes emphasize the significance of optimizing hyperparameters in Machine Learning models to enhance model performance.

**Table X. Predictions of 10 molecules from the validation set for both models**

| **Molecule ChEMBL id** | **Activity (true)** | **Random Forest prediction** | **XGBoost prediction** |
| --- | --- | --- | --- |
| **CHEMBL142764** | 0 | 0 | 0 |
| **CHEMBL1834330** | 1 | 0 | 1 |
| **CHEMBL564290** | 0 | 0 | 1 |
| **CHEMBL3401128** | 1 | 1 | 1 |
| **CHEMBL3398202** | 1 | 0 | 1 |
| **CHEMBL3398199** | 0 | 1 | 1 |
| **CHEMBL4753139** | 1 | 1 | 1 |
| **CHEMBL562814** | 0 | 0 | 0 |
| **CHEMBL296586** | 1 | 0 | 1 |
| **CHEMBL3410102** | 0 | 1 | 1 |

In conclusion, the CHEMO platform is designed to provide comprehensive and practical training in Python programming language, Dataframes, visualization, and Machine Learning algorithms as applicable in CADD research. The CHEMO platform can equip learners with practical knowledge, skills, and experience required in modern-day drug discovery and aid them in developing effective drug candidates.

The CHEMO platform provides learners with practical knowledge of applying machine learning algorithms to classify compounds in CADD research. Besides, learners will acquire skills on how to evaluate models performance, followed by hyperparameter tuning. This module aims to equip learners with practical knowledge of how Machine Learning can be applied to classify compounds in CADD research, evaluate models, and find the best hyperparameters settings. Additionally, learners will learn how to optimize and maximize the performance of the model.

# DISCUSSION

Computer-aided drug design (CADD) has become an essential aspect of the pharmaceutical industry, which aims to develop and design new therapeutic drugs. Many universities and training centers around the world are offering courses on CADD to train students in this emerging field. However, learning CADD can be challenging due to the complex concepts involved, and the practical exercises require a comprehensive understanding of scientific programming languages and tools. To address these challenges, a web-based teaching tool has been developed using Jupyter, Python, and scientific packages. The main objective of this tool is to facilitate the learning process of CADD, making it more interactive, accessible, and engaging for students.

Potential applications of the web-based teaching tool in various educational settings and contexts

Summary of the study's results and key takeaways

Reflection on the strengths and limitations of the tool's design and implementation

Discussion of the potential impact of the tool on the field of computer-aided drug design education

# CONCLUSION

Summary of the key findings and implications of the study

Recommendations for future research and development of web-based teaching tools for CADD

Final thoughts on the importance of effective teaching tools for computer-aided drug design education.